REMARKS

Claims 1-29 are pending in this application, of which Claim 25 is currently amended and Claims 26-29 are new. The specification is amended to correct a typographical error. The applicant respectfully requests entry of the amendments and consideration of Claims 1-29.

The Rejections

Claims 1-25 are rejected under 35 U.S.C. §102(b) as being anticipated by Geller et al. (U.S. Patent No. 5,844,554).

Regarding Claim 1:

Independent Claim 1 recites a method for performing a product configuration. The product configuration is associated with a configuration problem defining a number of constraints, one or more variables, and domain members associated with each variable. The method comprises receiving user input specifying at least one selected domain member, and producing a result that identifies incompatibilities between the domain members caused by the at least one selected domain member. Further, the method comprises modifying the result by detecting and eliminating incompatibilities caused solely by bounceback behavior.

Among other possible differences, the method taught by Geller et al. does not detect and eliminate incompatibilities in a result when those incompatibilities are caused solely by bounceback behavior.

The term "bounceback" is defined in the specification and in the art. One characterization of bounceback behavior was provided on page 4 of the communication filed March 7, 2003 (Amendment A). Specifically:

> Bounceback behavior is described in the specification by way of example (paragraphs 0005 and 0006) involving the configuration of a computer system. In the example, a set of constraints exists between processors and disk drives such that a 500 MHz processor is only compatible with 10 and 20 Gbyte disk drives, and a 700 MHz processor is only compatible with a 30 Gbyte disk drive. When the 500 MHz processor is selected in the example, constraint propagation causes the 30 Gbyte to be eliminated, and the elimination of the 30 Gbyte disk drive further causes the 700 MHz processor to be eliminated. Thus, *bounceback behavior occurs where constraint propagation causes non-selected domain members of a variable to be eliminated in creating a result because another domain member of the same variable has been selected.* (Italics added for emphasis.)

The Examiner is specifically directed to the sections shown above in italics. The Applicant is unable to find *any* discussion of bounceback behavior in the text cited by the examiner on page 3 of the current office action, "(col. 10, lines 64 – col. 11, line 27; col. 12 lines 51-56; col. 18, line 26 – col. 19, line 9; col. 23, line 46 – col. 24, line 47; col. 24, lines 59 – col. 25, line 52; col. 25, line 64 – col. 26, line 44; col. 28, line 56 – col. 29, line 32)." The Applicant respectfully requests that the Examiner more specifically point out how Geller et al. teaches "detect[ing] and eliminat[ing] incompatibilities in a result when those incompatibilities are caused solely by bounceback behavior" as specified in Claim 1, or withdraw the rejection.

Further, Geller et al. does not detect incompatibilities created by bounceback behavior. In fact, Geller et al. avoids the possibility of bounceback by severely limiting their approach to product configuration. Specifically, Geller et al. requires that dependencies be structured in a simple hierarchical relationship (Col. 4 lines 40-46, Col. 18 lines 42-48, Fig. 16A, and Col. 29 lines 5-7) and that restraints be resolved according to this relationship (Col. 37 lines 17-22) when the program is *compiled.* Thus, each variable must be determined by a user in an *order* that depends on the hierarchy. This disadvantageous approach avoids bounceback at a significant cost in utility and teaches

away from the current invention. For example, in Geller et al. the order in which a user could select a processor speed and disk drive size would be fixed. In contrast, the invention allows bounceback to occur and then "detect[s] and eliminate[s] incompatibilities in a result when those incompatibilities are caused solely by bounceback behavior (Claim 1). Because the invention allows bounceback, variables may be specified by a user in a variety of sequences or orders.

Geller et al. specifically teaches that "the computation ***cannot*** be accurately computed" unless calculated in a specific order (Col 37 lines 28-29, font changed for emphasis). This teaching is in direct contradiction to the claimed invention which accurately generates a result regardless of the order in which variables are specified. In the invention variables can be specified in a variety of orders because methods of the invention may include modifying the result by detecting and eliminating incompatibilities caused solely by bounceback behavior. It is the position of the Applicant that, rather than anticipating Claims 1-25, Geller et al. specifically and directly teaches away from the invention. The Applicant, therefore requests that the rejections be withdrawn.

Because Geller et al. teaches away from a system that would include bounceback behavior, Geller et al. cannot teach *detection* of incompatibilities that are caused by bounceback behavior, as recited in Claim 1. The Applicant requests that the Examiner specifically point out a teaching of Geller et al. that specifically anticipates *detection* of incompatibilities that are caused by bounceback behavior, or withdraw the rejection.

Furthermore, because Geller et al. does not teach detection of incompatibilities that are caused by bounceback behavior, Geller et al. cannot teach detection of incompatibilities that are caused *solely* by bounceback behavior, as recited in Claim 1.

The Applicant requests that the Examiner specifically point out a teaching of Geller et al. that specifically anticipates detection of incompatibilities that are caused *solely* by bounceback behavior, or withdraw the rejection.

Furthermore, because Geller et al., does not teach detection of incompatibilities that are caused by bounceback behavior, Geller et al. cannot teach *elimination* of incompatibilities that are caused by bounceback behavior, as recited in Claim 1. The Applicant requests that the Examiner specifically point out a teaching of Geller et al. that specifically anticipates *elimination* of detected incompatibilities that are caused by bounceback behavior, or withdraw the rejection.

Furthermore, because Geller et al., does not teach detection of incompatibilities that are caused by bounceback behavior, Geller et al. cannot teach *modification of a result* based on bounceback behavior, as recited in Claim 1. The Applicant requests that the Examiner specifically point out a teaching of Geller et al. that specifically anticipates *modification of a result* based on bounceback behavior, or withdraw the rejection.

For at least the several reasons provided, independent Claim 1 is not anticipated by Geller et al. The Applicant therefore requests that the rejection be withdrawn.

Regarding Claims 2-4:

Claims 2-4 are dependent on Claim 1 and are, therefore, novel for at least the same reasons as Claim 1; on these bases the Applicant requests that the rejection of Claims 2-4 be withdrawn.

Further, the Applicant notes that Claim 2 is additionally novel in that it recites generating a configuration page based on the modified result so that domain members identified as being incompatible due to bounceback behavior are not marked as conflicted

choices on the configuration pages. Because Geller et al. does not teach the *modified result* of Claim 1, Geller et al. cannot anticipate use of the modified result for generating a configuration page, as recited in Claim 2. The Applicant, therefore, requests that the rejection be withdrawn.

Furthermore, because Geller et al. does not teach *identification* of domain members as being incompatible *due to bounceback behavior*, Geller et al. cannot anticipate generating a configuration page on which incompatible domain members are *not* marked based on this criteria, as recited in Claim 2. The Applicant, therefore, requests that the rejection be withdrawn.

Regarding Claim 5:

Independent Claim 5 recites a system for performing a product configuration associated with a configuration problem defining a number of constraints, one or more variables, and domain members associated with each variable. The system comprises a configuration engine and a bounceback detection module. The configuration engine is adapted to receive user input specifying at least one selected domain member and to produce a result that identifies incompatibilities between the domain members caused by the at least one selected domain member. The bounceback detection module is adapted to modify the result by detecting and eliminating incompatibilities caused solely by bounceback behavior.

As noted with respect to Claim 1, since Geller et al. specifically teaches away from a system that includes bounceback, Geller et al. cannot teach any of the several limitations of Claim 5 associated with bounceback. Specifically, Geller et al. cannot anticipate "*a bounceback detection module* operatively coupled to the configuration

engine, the bounceback detection module adapted to *modify* the *result* by *detecting* and *eliminating* incompatibilities caused *solely* by bounceback behavior," as recited in Claim 5 (font changed for emphasis). The Applicant requests that the Examiner specifically point out teachings of Geller et al. that anticipate each and every limitation arising from the phrases "a bounceback detection module," "modify," "the result," "detecting," "eliminating," and "solely" as recited and used in Claim 5. Again, the Applicant is unable to find any teachings relating to bounceback in the specific columns and lines cited by the Examiner. The Applicant requests that the Examiner specifically point out how the text cited on page 3 of the current office action relates to bounceback and to the limitations of Claim 5, or that the Examiner allow Claim 5.

Regarding Claims 6 and 7:

Claims 6 and 7, depending from Claim 5, are novel over Geller et al. for the same reasons as Claim 5. Applicant, therefore, requests that the Examiner withdraw the rejections of Claims 6 and 7.

Furthermore, Claim 6 is additionally novel in that it recites "a page generation module operatively coupled to the configuration engine, the page generation module adapted to generate a configuration page based on the modified result so that domain members identified as being incompatible due to bounceback behavior are not marked as conflicted choices on the configuration page, and to provide the configuration to the user." Because Geller et al. does not teach the *modified result* of Claim 5, Geller et al. cannot anticipate use of the modified result for generating a configuration page, as recited in Claim 6. The Applicant, therefore, requests that the rejection be withdrawn.

Furthermore, because Geller et al. does not teach *identification* of domain members as being incompatible *due to bounceback behavior*, Geller et al. cannot anticipate generating a configuration page on which incompatible domain members are *not* marked based on this criteria, as recited in Claim 6. The Applicant, therefore, requests that the rejection be withdrawn.

Regarding Claim 8:

Independent Claim 8 recites a method that comprises at least the limitations of Claims 1 and 2 discussed above. Thus, for the reasons provided above with respect to Claims 1 and 2, Claim 8 is also novel over Geller et al. For at least these reasons, Applicant requests that the Examiner withdraw the rejection of Claim 8.

Regarding Claim 9:

Independent Claim 9 recites a method for detecting bounceback behavior associated with a configuration problem defining a number of constraints, one or more variables, and domain members associated with each variable. The method comprises: receiving a domain member selection for a particular variable; setting a bounceback detection bit vector associated with each non-selected domain member of the particular variable so that each of those bounceback detection bit vectors indicates bounceback behavior; setting an elimination flag associated with each non-selected domain member of the particular variable so that each of those elimination flags indicates that its associated domain member is tentatively eliminated; propagating the constraints to identify eliminated domain members of the variables; setting the bounceback detection bit vector of the eliminated domain members to indicate which variable caused their elimination; and setting the elimination flag of each of the eliminated members.

The Applicant respectfully traverses the Examiner's statement that "Geller discloses a method for detecting bounceback behavior associated with a configuration problem (col. 11, lines 3-27; col. 23 line 46 – col. 24, line 47)." In fact, the Applicant fails to see *any* teaching within the cited text relating to bounceback behavior. The Applicant, therefore, requests that the Examiner specifically explain how col. 11, lines 3-27, and col. 23 line 46 – col. 24, line 47 teach "*a method for detecting bounceback behavior*" as recited in Claim 9, or that the Examiner allow this claim.

Further, the Examiner "has interpreted bit vector as parameter" (bottom of page 3 in the current office action) when, in fact, it previously was pointed out in the remarks associated with Amendment A (page 4) that the bounceback detection bit vector includes a variety of features not found in a general parameter, such as that taught by Geller et al. Specifically, the Applicant has previously stated:

> In the present invention, *a result can be a bounceback detection bit vector* (page 11 line 1), and "[e]ach member of each variable included in a given configuration problem is associated with ... a bounceback detection bit vector" (page 10 line 22 – page 11 line 1). "*[E]ach bit position in a bounceback detection bit vector corresponds to a particular variable*" (page 11 lines 10-11). When a user selects a first member from a variable, if constraint propagation indicates that a second member is incompatible with the first member, *the second member will have a bit set in its bounceback detection bit vector*, and the bit that is set will be in a position that represents the variable of the first member. Thus, if a member is of a particular variable, and a bit for that same variable is set in the bounceback detection bit vector for that member, *then the result identifies a bounceback. If no other bits are set in that bounceback detection bit vector, then the incompatibility is due solely to bounceback.* Accordingly, the present invention is able to evaluate the result for each member to detect whether an incompatibility for that member is due solely to bounceback behavior. (Italics added for emphasis.)

The Examiner is specifically directed to the sections shown above in italics. In view of these characterizations of a bounceback detection bit vector, the Applicant is not able to discern any teachings in the cited art that anticipate a bounceback detection bit

vector as recited in Claim 9. The Applicant requests that the Examiner cite further support for the Examiner's interpretation of "bounceback detection bit vector" or withdraw the rejections associated with Claim 9.

As noted above, Geller et al. does not address bounceback detection. Accordingly, Geller et al. cannot teach setting a bounceback detection bit vector associated with each non-selected domain member so that each of the bounceback detection bit vectors indicates bounceback behavior. Likewise, Geller et al. does not teach setting an elimination flag associated with each non-selected domain member of the particular variable so that each of those elimination flags indicates that its associated domain member is tentatively eliminated. More particularly, Geller et al. does not disclose that a domain member can be *tentatively* eliminated. Additionally, Geller et al. does not teach setting a bounceback detection bit vector of an eliminated domain member to indicate which variable caused its elimination. Although, a bit in a parameter of Geller et al. can indicate that a first element has been removed from a configuration, there is nothing in Geller et al. that correlates that bit to a second element whose selection caused the removal of the first element. Lastly, Geller et al. does not teach setting an elimination flag of each of the eliminated members. For at least these reasons Claim 9 is not anticipated by Geller et al. The Applicant, therefore, requests that Claim 9 be allowed.

Regarding Claims 10-20:

Claims 10-20 are dependent from Claim 9 and Applicant, therefore, requests that Claims 10-20 be allowed for at least the same reasons as Claim 9.

Claim 10 further recites initializing the bounceback detection bit vector for each domain member of each variable, and initializing the elimination flag for each domain

member of each variable. Because the bounceback detection bit vector of Claim 9 is not taught by Geller et al., Geller et al. cannot teach initialization of the bounceback detection bit vector. Further, because the elimination flag is not taught by Geller et al., Geller et al. cannot teach initializing the elimination flag. For these further reasons, the Applicant requests that the rejection of Claim 10 be withdrawn.

Claim 12 further recites "wherein bounceback detection bit vectors that indicate bounceback behavior indicate that the particular variable associated with the selected domain member is responsible for elimination of the non-selected domain members." Applicant is not able to find teachings, within the citations provided on page 5 of the current office action, of anything that indicates that *the particular variable associated with the selected domain* member is *responsible* for *elimination* of the *non-selected domain members*. The Applicant respectfully requests that the Examiner more specifically point out this teaching or allow Claim 12.

Claim 13 further recites "confirming the tentative elimination of a non-selected domain member in response to the bounceback detection bit vector associated with that non-selected domain member not indicating bounceback behavior as a result of subsequent constraint propagation." Applicant is not able to find teachings, within the citations provided on page 5 of the current office action, of data "not indicating bounceback behavior as a result of subsequent constraint propagation," much less using "a bounceback detection bit vector associated with that non-selected domain member" as that data, as specified in Claim 13. The Applicant respectfully requests that the Examiner more specifically point out this teaching in the cited text of Geller et al. or allow Claim 13.

Claim 14 further recites overriding the tentative elimination of a non-selected domain member in response to the bounceback detection bit vector associated with that non-selected domain member indicating bounceback behavior despite subsequent constraint propagation. Applicant is not able to find teachings, within the citations provided on page 5 of the current office action, of "*overriding* the tentative elimination of a non-selected domain member in response to the bounceback detection bit vector." The Applicant respectfully requests that the Examiner more specifically point out this teaching or allow Claim 14.

Claim 15 further recites "wherein the step of setting the bounceback detection bit vector of an eliminated domain member to indicate which variable caused that domain member's elimination includes: based on the constraints, identifying a domain member causing the eliminated domain member to be eliminated; and copying the bounceback detection bit vector associated with the identified domain member to the bounceback detection bit vector associated with the eliminated domain member." Applicant is not able to find teachings, within the citations provided on page 5 or 6 of the current office action, of "*copying* the bounceback detection bit vector associated with the identified domain member to the *bounceback detection bit vector associated* with the eliminated domain member." The Applicant requests that the Examiner more specifically point out this teaching or allow Claim 15.

Claim 16 further recites "wherein the step of setting the bounceback detection bit vector of an eliminated domain member to indicate which variable caused that domain member's elimination includes: based on the constraints, identifying a join corresponding to a disjunction; logically ANDing the bounceback detection bit vectors associated with

the domain members included in the join thereby producing a resulting bounceback detection bit vector; and copying the resulting bounceback detection bit vector to the bounceback detection bit vector associated with the eliminated domain member." Applicant is not able to find teachings, within the citations provided on page 5 or 6 of the current office action, of "*identifying a join* corresponding to a *disjunction,*" "logically *ANDing* the bounceback detection bit vectors." or "*copying the resulting bounceback detection bit vector* to the bounceback detection bit vector *associated with the eliminated domain member.*" The Applicant requests that the Examiner more specifically point out this teaching, with specific attention to each of the phrases shown in italics, or allow Claim 16. Note that the ANDing operation is with respect to the bounceback detection bit vectors and not with respect to constraints.

Claim 17 further recites "wherein the step of setting the bounceback detection bit vector of an eliminated domain member to indicate which variable caused that domain member's elimination includes: based on the constraints, identifying a join corresponding to a conjunction; logically ORing the bounceback detection bit vectors associated with the domain members included in the join thereby producing a resulting bounceback detection bit vector; and copying the resulting bounceback detection bit vector to the bounceback detection bit vector associated with the eliminated domain member." Applicant is not able to find teachings, within the citations provided on page 5 or 6 of the current office action, of "identifying a join corresponding to a *conjunction*" or "logically *ORing* the bounceback detection bit vectors," or "*copying* the resulting bounceback detection bit vector ..." The Applicant requests that the Examiner more specifically point out this teaching, with specific attention to each of the phrases shown in italics, or

allow Claim 17. Note that the ORing operation is with respect to the bounceback detection bit vectors, not with respect to constraints.

Claim 18 further recites "generating a configuration page based on the constraints so that domain members identified as being eliminated due to bounceback behavior are not marked as conflicted choices on the configuration page; and providing the configuration page to a user." Applicant is not able to find teachings, within the citations provided on page 6 of the current office action, of "domain members *identified* as being eliminated *due to bounceback behavior* are *not* marked." The Applicant requests that the Examiner more specifically point out this teaching, with specific attention to each of the phrases shown in italics, or allow Claim 18.

Regarding Claims 21-25:

Independent Claim 21 recites a method that comprises at least the limitations of Claims 1, 2, 9, and 10 discussed above. Thus, for the reasons provided above with respect to these claims, Claim 21 is also novel over Geller et al. For at least these reasons, Applicant requests that the Examiner withdraw the rejection of Claim 21 and Claims 22-25 which are dependent on Claim 21.

Claim 22 further recites "wherein the step of setting the bounceback detection bit vector of an eliminated domain member to indicate which variable caused that domain member's elimination includes: based on the constraints, identifying a domain member causing the eliminated domain member to be eliminated; and copying the bounceback detection bit vector associated with the identified domain member to the bounceback detection bit vector associated with the eliminated domain member." Applicant is not able to find teachings, within the citations provided on page 6 of the current office action,

of "*identifying* a domain member *causing* the eliminated domain member *to be eliminated*," or "*copying* the bounceback detection bit *vector associated*..." The Applicant requests that the Examiner more specifically point out this teaching, with specific attention to each of the phrases shown in italics, or allow Claim 22.

Claim 23 further recites limitations similar to those of Claim 16 and the points discussed above with regard to Claim 16 also apply to Claim 23. The Applicant requests that the Examiner more specifically point out teachings within the cited text that anticipate these limitations, or allow Claim 23.

Claim 24 further recites limitations similar to those of Claim 17 and the points discussed above with regard to Claim 17 also apply to Claim 24. The Applicant requests that the Examiner more specifically point out teachings within the cited text that anticipate these limitations, or allow Claim 24.

Claim Amendments

Claim 25 is amended to correct a typographical error.

New Claims

Regarding New Claims 26-29:

Support for New Claims 26-29 can be found within the specification as filed. For example, Claim 26 is supported by some of the same passages supporting Claims 1, 5, 8, 9 and 21.

Applicant believes that all pending claims are allowable and respectfully requests that the Examiner issue a Notice of Allowance. Should the Examiner have questions, the Applicant's undersigned representative may be reached at the number provided.

Respectfully submitted,

John M. Mela

Date: October 17, 2003

Steven M. Colby, Ph.D. Reg. No. 50,250
Carr & Ferrell *LLP*
2200 Geng Rd.
Palo Alto, CA 94303
Phone (650) 812-3400
Fax (650) 812-3444